



BUILDING & HOSTING SOLUTIONS FOR GOVERNMENT AGENCIES

US Department of Labor IT Modernization Strategy: From Vision to Implementation

WHITEPAPER

May 9, 2023

Solid State Operations, Inc

Contents

- Background 2
- Lock-In Case Study..... 4
- The Solution: Open and Modular 5
 - Definitions..... 6
 - Levels of Openness and Modularity 6
 - Level One: Architecturally Monolithic 6
 - Level Two: Walled Garden 7
 - Level Three: Proprietary Ecosystem 7
 - Level Four: Software Platform..... 8
 - Level Five: Competitive Marketplace 8
- General Recommendations 10
 - Fix the root cause and the symptoms will resolve 10
 - Focus resources and effort on Levels 4 and 5 10
 - Don't be afraid to disrupt 10
- Specific Recommendations..... 11
 - Create or sanction the software and infrastructure..... 11
 - Create or sanction a Trusted Authority. 11
 - Remove barriers to procurement..... 12
 - Encourage the use of the Marketplace..... 13

This white paper interacts with the recently published [US Department of Labor IT Modernization Strategy](#), looking at it in the context of both technology in general, and Unemployment Insurance (UI) modernization in particular, over the past, present and future time horizons. The goal is to help the reader understand the Department's new approach as well as offer recommendations for how best to achieve the Department's goals with the resources it plans to make available to the State Workforce Agencies (SWAs).

Background

"The COVID-19 pandemic has made it abundantly clear that the nation's unemployment insurance system is inadequate to protect workers from financial devastation when they lose a job." - Senator Ron Wyden & Senator Michael Bennet

The UI program is in trouble. Its technological underpinnings are crumbling, and no amount of money seems to be able to shore it up. The public is angry; legislators are frustrated, and current thinking has not been able to provide a good solution. The system does not scale, people are wrongly denied benefits or do not get benefits because of inefficient and slow adjudication processes - but at the same time fraud is rampant with criminal rings defrauding the system of billions of dollars.

And the problems are not just because of the pandemic – the pandemic simply highlighted structural problems that have been there for years, and that, unless we can provide revolutionary change, will continue after the pandemic.

In the 1980s the computer revolution hit UI. Mainframe systems began processing claims and taxes. The mainframe processing systems were a big step forward, but they still required a lot of manual work, and they were not accessible by the claimants. As the years went by, the public's expectations were shaped by the leaps and bounds of eCommerce. Convenience and self-service delivered over the internet became the norm; anything less became frustrating.

At the same time, technology progressed far beyond COBOL and flat files. It became very evident that in comparison to emerging technologies, mainframes are inflexible, difficult to work on, and expensive to maintain and operate. Many states have been desperate to find new options that can streamline business processes, reduce costs, and provide better security and privacy protocols for the massive amount of data maintained by the UI system.

The first response to the problems posed by the aging mainframe infrastructure is to propose generic "Modernization."

At first, this seems simple and sensible. Just replace the mainframe with web-based systems. And all you have to do is just follow industry: industry is moving to web-based systems, and they seem to be having good success. SWA's should adopt the same technology and the same techniques for implementing. The Agency can hire some of the big, successful technology companies to perform this modernization, and at the end of the day, these companies will deliver a web-based system that will replace the mainframe with a modern, user-friendly system that the claimants, employers, and staff will enjoy using.

The issue with this approach, however, is that besides being more complicated than everyone assumed, what they missed is that there are barriers to entry and barriers to procurement in government that do not exist in private industry. In fact, “Modernization” requires State Agencies to perform a complicated multi-year dance just to successfully navigate the state procurement processes.

There is no sugar-coating the fact that, with few exceptions, Modernization has not been successful.

“To date, fewer than half of states have modernized their unemployment benefits.... As of 2016, 26 percent of projects had failed and been discarded; 38 percent were past due, over budget, or lacking critical features and requirements; and 13 percent were still in progress...”

The story these numbers tell is quite discouraging: More than half the time, the process of acquiring the software is so expensive and onerous, and the risk of failure so high, that states would rather hold on to the antiquated systems. When they do embark on software acquisition, more than a quarter of the time, it is a complete waste and everything that they paid for is thrown away. Over a third of the time, while something is delivered, the project ends in some type of disappointment, recrimination, lawsuit, or congressional investigation. Only 23 percent of projects were unequivocally, even minimally successful. This is a disheartening percent, but honestly, not out of line with other government projects: across the board, only 13% of large government software projects are successful.ⁱ

Now, almost twenty years into the UI modernization movement, we’re in a good position to evaluate and course-correct. Below are some things that DID NOT work for states:

Every state for themselves: States largely had to fend for themselves with too many options and no overarching architecture guidance. .NET, Java, waterfall, agile, custom, COTS? Intuitively, we know that having 53 ways of doing things leads to inefficiency. In the mainframe era the GUIDE systems provided some states with a common starting point and promoted reusability and some level of standardization. Surveying the state of modernization, it is apparent that too often states and vendors have had little success in creating excellent web-based services. Instead, they have consistently fallen back to 1990’s era proprietary, monolithic solutions that do not communicate easily with outside systems and do not promote service reuse.

Relying on Agency staff to design, develop and deliver software: It should not be controversial to state that UI Agencies should focus their time, effort, and manpower on collection of UI Taxes and delivery of UI Benefits - not software design, development, and delivery. Creating excellent web-based services is hard. Some of the best minds in the country are paid a lot of money to go work in Silicon Valley to create exactly these types of excellent web-based services. And many fail - even in Silicon Valley.

“All-or-nothing” state procurements: Monolithic UI software – and monolithic procurements – force states to purchase the entire UI system from one vendor. The inability to divide UI into component services creates the complicated and risky all-or-nothing approach where either a vendor can supply all of the modernized system, or they cannot supply any of it. While there are many companies that could supply pieces of the modernized system, there are only several companies that can do it all. This creates limited choices, stifles competition and prevents states from implementing the eminently sensible recommendations from GSA’s Technology Transformation Services’ [Derisking Guide](#). The Derisking

Guide advocates breaking up complex system modernization into modular contracts of limited dollar value. This is impossible in an all-or-nothing approach.

Multi-State Consortia: In the modernization era, the multi-state “consortium” has been promoted as a way to gain some measure of economy of scale by having multiple states use a similar software design. In theory the software is paid for and designed once and then used by several states. Consortia have been an interesting experiment but have been largely unsuccessful. Most of the consortia have received tens of millions of dollars in federal funding but have ended up aborted. The problem seems to be that designing a single system that will work for multiple states is too difficult. Having states agree on common processes and terminology and then program the system to fit the common process has been prohibitively difficult. Based on their poor track record, multi-state consortia do not seem to be the solution.

This is the situation in which DOL’s IT Modernization Strategy was introduced.

US Department of Labor’s IT Modernization Strategy identifies issues with the current state of affairs:

- “technology breakdowns, delays, and new attacks from fraudsters”
- “changes [that] can be prohibitively costly, time-consuming, and risky.”
- “outdated technology”
- “systems that are brittle and inflexible to new requirements”
- “high cost of adapting systems to meet changing needs”

The *Modernization Strategy* is clear that these issues are really symptoms of a larger problem: “how those systems are designed and managed... an ‘all or nothing’ choice: holding onto antiquated systems or embarking on a risky system overhaul”

DOL is absolutely correct. And this “all-or-nothing choice” should really be called “vendor lock-in.” Vendor lock-in describes the situation where due to any number of factors (such as software architecture, implementation choices, or procurement choices) the State cannot unilaterally replace a piece of a vendor’s system with an alternative from a different vendor or open-source without replacing the whole system (and replacing the vendor in the process). Vendor lock-in is a serious phenomenon that raises the Agency’s costs and limits the Agency’s ongoing choice in how their UI system is implemented.

Lock-In Case Study

What does vendor lock-in look like, and how expensive is it really? In this section we examine a recent [news story](#) concerning one state’s obligatory payment to a locked-in vendor:

“The Legislature’s Contract Review Committee on Thursday authorized a request from the state agency to appropriate \$20 million in federally allocated funds toward enhancements on a data station linked to the [UI claims] portal.

“[The vendor], has been awarded the contract for the data station upgrades. The company previously undertook the overhaul of [the state’s] obsolete unemployment insurance system at the height of the pandemic.

“[The state representative] said an ongoing relationship with [the vendor] was a necessity since the company owns the data station software that powers the unemployment insurance portal.

“[The vendor] representatives must perform any changes and additions to the existing system, according to information from the department.

“‘We’ve got a lot of fraud on the back end of the system,’ [the state representative] said of the rationale behind the new ancillary contract with [the vendor].

“‘While the state has an established agreement with [the vendor],’ [the state representative] said, ‘This is additional work and, therefore, an additional contract.’

“During deliberations around the new contract, [the state representative] confirmed the Department of Labor endured voluminous fraudulent unemployment insurance claims, particularly in the months following the onset of COVID-19 in March 2020.”

Here we see that the State believes they have no choice but to pay the vendor (the “ongoing relationship” being “a necessity”) since the vendor “owns the data station software that powers the unemployment insurance portal.” Evidently the State cannot replace pieces of their system, and it’s all-or-nothing. The same vendor that owns the back-end system must perform the work on the front-end claim portal.

While the story does not detail the exact nature and extent of the “enhancements,” it’s easy to believe that the \$20M the vendor extracted from the State is due in large part to vendor lock-in - especially when according to the Agency representative, the vendor in question, as owner of the back-end systems was, at least in some way, party to “a lot of fraud on the back end of the system” and “voluminous fraudulent unemployment insurance claims.” And the vendor has evidently not fixed their system in the three years since “the months following the onset of COVID-19 in March 2020.”

This is clearly a case of vendor lock-in, and while we don’t know all the specifics, what we are told doesn’t look good. It highlights the need for a different approach.

The Solution: Open and Modular

This paper would submit that the root cause of all the modernization problems identified in the *Modernization Strategy* (including vendor lock-in) is:

Lack of competition, caused by the absence of an open, modular marketplace.

This is what makes DOL’s prescription in their “Vision for IT Modernization” such a bulls-eye. DOL’s vision is for “a new ecosystem built around open and modular solutions that promote innovation, software reuse, and incremental – rather than all-or-nothing – approaches to modernization.” Bingo! They hit the nail on the head. They have correctly identified the way out of the modernization quagmire.

Since DOL identifies the creation of a new, open, modular ecosystem as the panacea, and since this paper agrees with their assessment, the next step is to examine the concepts of “openness” and “modularity” beginning with a simple definition of each.

Definitions

- **Openness:** The degree to which the SWA has a choice in how the system is implemented.
- **Modularity:** The degree to which the system can be decomposed into separate but inter-related parts.

Beyond the simple definitions, there are several things to note about these concepts:

First, we believe that openness and modularity are both non-binary. It is not simply the case that an ecosystem is either open or not – or modular or not. There are levels of both modularity and openness, from lesser to greater.

Second, we believe that openness is dependent on modularity. To the extent that a system is not modular, it cannot be open. The discussion on Levels of Openness and Modularity below will explain this further.

Third, it is tempting to conflate “open” with “open-source.” This is not entirely accurate. While open-source components certainly do contribute to openness, and we are convinced that they are necessary to a well-rounded ecosystem, they are not the only – or even necessarily highest expression of – implementation “choice” for the SWA. Open-source components are often inexpensive or free to acquire and allow the SWA to modify source code. These are great features, and certainly do contribute to “choice” – but so does having many vendors being able to provide cost-effective and competitive alternatives. We believe it is more accurate to conflate “openness” with “choice.” A perfectly open system removes barriers to competition which produces the greatest choice of alternatives for the Agency. Unimpeded competition drives lower prices, better features, greater responsiveness, and better service for the State, claimants, and employers.

Levels of Openness and Modularity

In this section, we identify five distinct systems that could be implemented by a State. Each of these systems, by its nature, provides its own degree of modularity and openness to the State. The systems are presented here in order from least beneficial to most beneficial. This isn’t to say that there aren’t other variations that could be implemented, but we feel these five are representative.

Level One: Architecturally Monolithic

In an architecturally monolithic implementation, most of the system’s functionality is concentrated in one or two large, Swiss-Army-knife applications. In the UI world, you may have a Tax app and a Benefits app that have all the respective functionality built into that application. Being “architecturally monolithic” doesn’t depend on whether the application is a desktop app or a web app, nor does it depend on whether it is owned by a vendor or is open source.

With an Architecturally Monolithic system, even small changes potentially affect the entire system. Furthermore, the entire system needs to be redeployed with every change. This essentially requires that the entire system needs to be retested for every deployment. Of course, in a rapidly changing environment, whole-system regression testing is not feasible for every change, so it doesn’t get done. This slows the rate of change possible within the system as well as raises the risk of bugs potentially leading to anything from bad customer experiences all the way to data corruption.

Architecturally monolithic systems typically have a lower cost of acquisition (i.e., the cost to build and deploy it), but a higher cost of maintenance and cost of replacement.

- **Modularity:** Since monolithic is the opposite of “modular,” there is no modularity in this system.
- **Openness:** The state is “locked-in” to the vendor because they cannot upgrade or replace one part of the system without replacing the whole thing. The state is at the mercy of the vendor with regard to the cost and timing of system upgrades.

Level Two: Walled Garden

The Walled Garden describes an architecturally modular system owned by a vendor and effectively closed off to all others. A modular architecture breaks up the functionality of the system into discrete, self-contained units of functionality. Each of these units have their own concerns, separated from the rest of the system by application programming interfaces (APIs). The nature of the Walled Garden is that the vendor may allow limited access to the State or to other vendors, but the core system – including the APIs - is generally inaccessible.

- **Modularity:** The architectural modularity of the Walled Garden approach offers some functional improvements over the Architectural Monolith. It allows a change in one module to not affect any other module. As long as the API stays the same, changes only require the testing and redeployment of the module in which they were made. This reduces deployment risk, and - **theoretically** – increases the flexibility and rate of change of the system.
- **Openness:** The Walled Garden **does not** alleviate vendor lock-in. The fact that these APIs are not published or made available means that the system is effectively no different than a monolith. The vendor owns the entire system, and the state still cannot upgrade or replace one part of the system without replacing the whole thing. The State is still entirely at the mercy of the vendor with regard to the cost and timing of system upgrades.

Creating - or perpetuating - a Walled Garden approach is a choice made by the vendor and ratified by the State. The vendor *could* open their APIs (thereby creating a Proprietary Ecosystem, see below), but they choose not to. And the State, through their procurement of such a system, lends their approval.

Level Three: Proprietary Ecosystem

The Proprietary Ecosystem describes a modular system which allows access to the API by the State and other vendors. Like the Walled Garden, the Proprietary Ecosystem has a modular architecture which breaks up the functionality of the system into discrete, self-contained units of functionality. However, unlike the Walled Garden, the system **does** provide unrestricted access for the State and other vendors to the underlying API.

- **Modularity:** Architectural modularity offers the same advantages as the Walled Garden in terms of reduced deployment risk, and increased flexibility and rate of change of the system.
- **Openness:** The Proprietary Ecosystem **partially** alleviates vendor lock-in. Whether the State or the vendor owns the system, the hallmark of the Proprietary Ecosystem is that the Agency and vendors do have access to the API. This allows the State to substitute other vendors’ products if the other vendors can implement the API. This is good, but it is limited. The limitation is that the API on which the ecosystem is based is basically a “one-off,” meaning that the API only appears in the state(s) where the ecosystem is deployed. This is usually one, or at most, a handful of states. The problem is that while existing vendors might be motivated to retrofit

their existing products to work in the ecosystem, most potential vendors would **not** invest the effort to create new products for just one or two states. Without a significant vendor pool, there won't be true competition or true "choice" for the agency.

Level Four: Software Platform

Instead of a one-off vendor (or state) controlled API as in the Proprietary Ecosystem, a Software Platform is based on centralized interface standards called Governance. This Governance is managed by a "Trusted Authority," is documented and made available to all, and is designed for use in each state's implementation. This is similar to the way the ubiquitous internet standards are governed by the [World Wide Web Consortium](#).

In contradistinction to the previous approaches, a Software Platform is not really a "product" *per se*. The goal of a Software Platform is to create competition by bringing together and enabling transactions between disparate buyers and sellers – without being a buyer or seller itself. Think eBay or Airbnb.

With universal governance, vendors can create applications knowing that their work can be deployed easily in any state. Similarly, states can easily transfer applications from other states – applications simply "drop in" because they adhere to the same governance. This is what enables the "reusable solutions" advocated by the *Modernization Strategy*.

With the assurance of this type of governance, the risk-reward equation becomes much more favorable, and many new vendors will be willing to invest the effort to create new products. When that happens, there will be meaningful competition.

The Software Platform offers improvements to both modularity and openness:

- **Modularity:** Like the Proprietary Ecosystem and Walled Garden, the Software Platform incorporates architectural modularity. However, unlike the other two scenarios, the interfaces of the Software Platform are established and maintained by the Trusted Authority rather than a single vendor. This implies that changes to the interface standards will ostensibly be optimized as to their timing and effects for both the Agencies and all vendors, not just the one vendor who owns the API. While the real-world outcomes will depend on the competence, motives, and motivation of the Trusted Authority, this approach offers a significant theoretical improvement in modularity over the others.
- **Openness:** The Software Platform encourages a large vendor pool through its universal governance. The governance is ostensibly optimized for vendor participation by the Trusted Authority. This removes the all-or-nothing dilemma by enabling vendors to implement smaller modules of functionality. This removes barriers to entry for many new vendors, encourages more competition, and results in greater choice for the Agency.

These advances are good, but there is still room for improvement.

Level Five: Competitive Marketplace

To provide optimal "openness" – or competition or choice for the Agency, the solution must minimize barriers to entry for vendors as well as barriers to procurement for the State Agencies. The Software Platform removes many barriers to entry for vendors by allowing them to interact with a published, optimized API and implement small units of functionality rather than an entire UI system. However, the

Software Platform does not remove the barriers to procurement for State Agencies, nor the vendor barriers to entry related to procurement.

What we mean is that even if all our plans so far have worked perfectly, and there are dozens of vendor alternatives available - if the state Agency cannot procure these alternatives easily, in a timely manner - or even at all - those alternatives may as well not exist, and we have effectively done nothing to create an “open” system for the State. “Choice” for the State only exists when the State can quickly and easily replace one vendor’s implementation of a module with another vendor’s.

DOL wisely recognizes this. The *Modernization Strategy* says, “Effective modernization involves more than just technology. It also requires a new mindset and approach to building and buying technology that centers on modernization as a continual process and the belief that IT systems are never complete.” This is another bull’s eye from DOL. It is where the Competitive Marketplace introduces the final pieces to the puzzle.

Apple’s App Store is a great example of the Competitive Marketplace principles of lowering barriers to entry and barriers to procurement. The App Store provides one place for vendors to advertise, review, and sell their applications. Small vendors can be on an equal footing with big ones. Barriers to entry are lowered dramatically. But on the other side, barriers to procurement are also lowered. The App Store is a trusted payment implementor. The store provides a facility for customer reviews. Additionally, the App Store tests and certifies each application for fitness to be sold within the store. Each of these things make it much easier for the buyer to purchase the application by removing financial, quality, and security concerns.

The Competitive Marketplace incorporates the tools necessary to allow the state to quickly and easily replace one vendor’s implementation of a module with another vendor’s. These include:

- **Certification Criteria:** A Trusted Authority provides criteria necessary to achieve in order to be certified as a legitimate vendor in the marketplace. If a vendor’s module is tested to meet these criteria, it is certified and made available in the Marketplace.
- **Store:** The Store provides the ability to the State to search for, learn about, purchase and begin using any of the certified vendor modules with a few button-clicks. Again, the App Store is a good analog.

These tools must obviate the need for a lengthy, expensive, and choice-killing RFP. If a module is certified for the Marketplace, and available to the State, the State should be able to procure it and begin using it with a click of a button.

- **Modularity:** The Competitive Marketplace maintains the highest standards of modularity as embodied in the Software Platform approach.
- **Openness:** The Competitive Marketplace exceeds the openness of the Software Platform by further reducing barriers to entry and barriers to procurement. A large vendor pool coupled with instant procurement ability is the apex of openness. It is the only real way to create “a new mindset and approach to building and buying technology that centers on modernization as a continual process.”

General Recommendations

This section goes over philosophical or conceptual recommendations that we believe should form the mentality of coming guidance and funding. Achieving the goals in the *Modernization Strategy* is absolutely worthwhile but will not be easy. The strategy is a paradigm shift, and the goals are technologically difficult and potentially disruptive. Continuously maintaining the mindset and philosophy explicated in this section will be crucial to completing the mission.

Fix the root cause and the symptoms will resolve

We have to focus on root causes and not be satisfied until the root causes are fixed. Preoccupation with fixing symptoms is a waste of time. The symptoms are poor products: high prices, inadequate features, unreliability, slow turnaround, poor service, etc. The root cause is lack of competition. When there are only a handful of vendors that can implement a complete system, there's not much competition at the time of procurement. But what's even worse is that when a locked-in vendor, year after year, faces no challenge to their incumbency, there is no competition for decades.

The way to introduce competition is through a Marketplace with published, modular standards curated by a Trusted Authority; where a state Agency can substitute module alternatives at will. This increases the size of the vendor pool in the following ways: Modular standards allow vendors to deploy smaller units of functionality rather than all-or-nothing; universal standards assure vendors of a wider marketplace for their products than a simple one-off; and removing the barriers gives vendors confidence that state agencies can actually procure their products and thus they can achieve a rapid return on investment.

All the rest will follow from competition: Lower price, better features, more reliability, faster turnaround, better service.

Focus resources and effort on Levels 4 and 5

Levels 1, 2, and 3 are dead ends. Moving a state, for example, from an Architectural Monolith to a Walled Garden isn't really that useful. Sure, they may have some better builds and deployments, but we haven't really done anything to fix the underlying structural problems that limit reusability and competition. Those are only fixed with a Software Platform leading to a Competitive Marketplace.

In 2023, the confluence of awareness, funding, and technology gives us an occasion to take a giant step towards fixing the modernization problem through the creation of an open, competitive marketplace. USDOL cannot let this once-in-a-decade opportunity pass without standing up this marketplace - with critical mass to take off. USDOL cannot squander this chance by getting distracted by Levels 1, 2, and 3 rather than spending its effort on Levels 4 and 5.

Don't be afraid to disrupt

There will be winners and losers in this transition. It may not yet be apparent who they are, but we do know that there are entrenched interests for whom the existing barriers to entry and barriers to procurement are an integral part of their current business model. They can adapt to a marketplace business model, but if they choose not to, they may create headwinds.

Specific Recommendations

If lack of competition is the root cause, let's review the "why's" to figure out what specific things we can do to fix it:

- I. Why is there no competition?
 - 1) Lack of a Marketplace.
- II. Why is there no Marketplace?
 - 1) Lack of a Software Platform.
 - 2) No Critical Mass
 - 3) State barriers to Procurement.**
- III. Why is there no Software Platform?
 - 1) No procurement of software/infrastructure.**
 - 2) Lack of a Trusted Authority.**
- IV. Why is there no Critical Mass
 - 1) No authoritative mandate**

The items in bold are the lowest-level items that we can do right now. If we do these things, we can create the competition we need. These items, then, form this section's Specific Recommendations.

There are only five specific recommendations, but we believe each of them are vital to achieving the goals laid out in the *Modernization Strategy*. These five recommendations are not "nice-to-haves" or things that should maybe be done "down the road." They are things that, if not accomplished with this round of guidance and funding, will neuter – and ultimately kill - the *2023 Modernization Strategy*.

Create or sanction the software and infrastructure

The software and infrastructure are necessary to form the backbone of the Software Platform and then eventually the Marketplace. For this Platform to be universally accepted, the authority for its establishment must come from the national level.

- 1. As part of the Modernization Strategy, USDOL needs to create or sanction the necessary software and infrastructure – as well as the ability to deploy, maintain, audit, recover, and improve the platform over time.**

Create or sanction a Trusted Authority.

The Trusted Authority is a hallmark of the Marketplace. The Marketplace does not function properly without a Trusted Authority. The Trusted Authority brings standardization, credibility, fairness, and efficiency to the Marketplace. Without a Trusted Authority, it is just a Software Platform.

We believe the Trusted Authority is the embodiment of what the *Modernization Strategy* describes as "providing foundational perspectives around 'what good looks like'".

The Trusted Authority should be 1) trusted by the stakeholders in the UI community. They must not have even the appearance of questionable motives. And the Trusted Authority must be 2) an authority on UI software, both the business rules, and the technology.

Like the W3 Consortium, the Trusted Authority can incorporate vendors, customers, government officials, and other stakeholders to accomplish their mandates – but someone needs to be in charge, and they should be sanctioned by USDOL. This latter part is important.

2. As part of the *Modernization Strategy*, USDOL needs to create or sanction a Trusted Authority which meets the criteria above.

The Trusted Authority should be charged with establishing Governance. Governance is the interface standards between modules on the Platform. Governance is absolutely necessary to make the Software Platform work correctly. And stable, fair, competent governance should be characteristic of the Marketplace. The Trusted Authority should have final say over what these interfaces look like. Prudence would likely dictate that the Trusted Authority bring in a wide range of stakeholders to workshop the standards, but at the end of the day, the Trusted Authority is accountable for the final product.

Finally, the Trusted Authority should be charged with establishing marketplace certification criteria. It is very important to determine what makes a product fit for the Marketplace. This is a special, indispensable example of defining “what good looks like.” “Good” must be defined across as many dimensions as practicable: functionality, performance, security, accessibility, interoperability, etc. Once these dimensions are defined, they can be tested, and vendor services can be approved or certified. This provides a well-defined target for vendors to measure their products against and protects the state by ensuring that it is getting a good product. Further, the existence of this standard may allow for streamlining of procurement (see below).

Remove barriers to procurement.

Barriers to procurement must be identified and mitigated. The goal is to provide competition in the form of the State’s ability to immediately replace one vendor’s module implementation with another vendor’s implementation. The ability to do this is what opens competition in the Marketplace. When incumbents face credible threat of replacement, they stay motivated to provide the best products and services for their customers. That is the *sine qua non* of competition.

The specific barriers to procurement may be different in each state, but there are some common challenges. For example, state procurement typically relies on goods and services being well-defined commodities, easily compared on common characteristics and price. For instance, if a state wants to buy tires for their police cruisers, they can specify a precise, well understood specification of “275/60VR17 M+S” that multiple vendors could quickly meet. Typically, the company that supplies these exact tires at the lowest price will win the contract. That works for tires.

Unfortunately, states generally must often use the same process when buying software. The process does not work very well for software. First, it requires the state to create a list of desired features to evaluate different products. For UI procurements, this list of features can be several thousand entries. The state must come up with precise definitions and ways to compare products on each feature. States often spend years writing procurement requirements and evaluating the vendors against criteria the state may not understand.

One solution with precedence is for USDOL to establish a blanket purchase order for states to both get into the Marketplace and acquire any content modules that have been tested to meet the Trusted Authority’s certification requirements. Such an approach would protect the states while at the same time opening the Marketplace to immediate qualified vendor selection.

3. USDOL must take the steps necessary to allow States to make immediate module replacements.

Encourage the use of the Marketplace

To build a Marketplace characterized by inter-state product interoperability leading to vendor participation, growth of the vendor pool, competition, and economies of scale, USDOL needs to establish a Critical Mass of states and vendors that are using the Marketplace such that it becomes the *de facto* standard for modernization going forward. Critical Mass is important because if the Marketplace is just a “one-off,” we haven’t really progressed beyond the Proprietary Ecosystem. USDOL needs to take steps to ensure that the standard promulgated by the Trusted Authority becomes ubiquitous across both states and vendors.

Again, an analogy can be found with the W3 Consortium. In the early 2000s each web browser had its own standard for how it would interpret HTML. Web development was difficult because of having to account for many different standards. In reality, most web developers only supported the top two or three browser implementations. The W3C attempted to standardize the HTML interfaces so that any browser that implemented the standards could be used to reliably view a web page. However, web developers could only rely on the W3C standard once it was adopted by a critical mass of the browsers. Until then, it was just pie-in-the-sky.

How can USDOL promote ubiquity of the Trusted Authority’s standards? The nature of the UI State-Federal Partnership does not lend itself to an “authoritative mandate” to use the Marketplace. Rather we advocate for “encouragement” to use it. The goal is to build that critical mass of states using the marketplace.

State encouragement may take the form of financial incentive, special training, prestige-enhancing recognition, streamlined procurement, etc.

Our initial estimate is that somewhere around 5 states should constitute critical mass. After that, joining the Marketplace becomes very easy for each additional state. Among other benefits at that level would be: precedent, tooling, a selection of content modules, a user community, and DOL support.

4. As part of this round of guidance and funding, USDOL needs to ensure that at least 5 pilot states are using the Marketplace.

Additionally, USDOL needs to ensure that there is a selection of vendor products available on the Marketplace. Ideally this would consist of a mix of propriety modules as well as open-source content, but at minimum, there must be at least one implementation of every Governance available to the pilot states.

5. As part of this round of guidance and funding, USDOL needs to ensure that there are enough vendor and state resources ready and able to create or adapt all of the necessary content modules for the pilot states.

ⁱ Projects valued at \$6M or greater, in Europe and the United States, that were completed satisfactorily, on time, and within budget. From The Standish Group's "Haze," based on their CHAOS database. Cited in De-Risking, pg. 7